PROFIT MAXIMIZING IN RIDE POOLING SYSTEMS WITH INDIVIDUALIST PRICING

A Thesis

Presented to the Faculty of the Graduate School

of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

M.S.

by

Yuchen Sun

August 2018

ABSTRACT

Shared vehicle systems like bike-sharing, car-sharing and ride sharing have become an essential part of the urban transit system. Ride-sharing services provided by companies like Uber or Lyft make it easier for those who are unable to operate a personal vehicle in big cities like NYC to find a ride with a lower price.

This study demonstrates the advantages of the ride sharing service compared to the ordinary non-shared ride service based on the simulation model of both systems. This study also illustrates a method to optimize the ride sharing systems by utilizing an individualist pricing strategy. An optimal discount for each ride sharing request to maximize the total expected profit of the system can be found based on the decision tree model and discrete choice model implemented in this study.

The purpose of the decision tree model is to determine whether a ride-sharing request can be shared with others in the system while the discrete choice model is used to estimate the passenger's willingness to accept a certain price for the ride-sharing service. These two factors are of vital importance when calculating the expected profit of serving a request.

The individual pricing strategy makes the price of the ride-sharing service more rational according to the features of the requests and thus can be an efficient way to attract more potential customers.

keyword: ride-sharing, pricing strategy

BIOGRAPHICAL SKETCH

Yuchen Sun is currently finishing his second year of studying in Civil & Environmental Engineering at Cornell University. In August 2018, he will graduate with a Master of Science degree in Transportation Systems Engineering.

Yuchen was graduate from Zhejiang University at Hangzhou, China in 2016. With strong interest in exploring intelligent transportation problems and the desperate desire to make the existed transportation systems more efficient. Yuchen decided to pursue his master's degree of Transportation Systems Engineering at Cornell University.

During his memorable years at Cornell University, Yuchen studied various skills that can be applied into solving intelligent transportation problems from the lectures provided by a number of programs and departments including Civil & Environmental Engineering, Operation Research & Information Science and Computer & Information Science.

The internship experience with the ride sharing R&D team of DIDI at the end of 2017 gave Yuchen a precious opportunity to get further understanding on how to commercialize the academic theories and made him resolved to explore the ride sharing pricing problem as the topic of thesis.

In addition to academics, sports like snowboarding or skydiving also makes Yuchen's time in Ithaca exciting. He's deeply impressed by the beauty of nature and the embracing culture while travelling around the United States. However, studying above Cayuga at Cornell is definitely the most precious memory in his life.

This document is dedicated to all Cornellians.

ACKNOWLEDGEMENTS

I would firstly appreciate the help and guidance of Professor Samitha Samaranayake of the Department of Civil & Environmental Engineering as my thesis advisor. His achievements in the area of intelligent transportation research encouraged me to explore the unknown and following my interests during my academic studies. Professor Samaranayake helped me whenever I ran into trouble and gave insightful comments on my thesis. I could not have the simulation model successfully utilized in this thesis without his guidance.

I would then appreciate the help of Professor Ricardo A. Daziano of the Department of Civil & Environmental Engineering as the minor advisor of the thesis. The lecture of Professor Daziano motivated me to build and implement the discrete choice model in my thesis. His inspiring personality also encouraged me to work through the difficulties in this study.

I would also acknowledge my thanks to the instructors of the lectures I've taken who have become my mentors during my two-year graduate study. They taught me to think critically and independently, which encourages me to explore further after my graduation.

Finally, I must express my profound gratitude to my parents and friends for always being there supporting me whenever I need their hand. I would never become who I am without them.

It is all of you that makes me proud of being a Cornellian. Thank you.

TABLE OF CONTENTS

CHAPTER 1

LITERATURE REVIEW AND INTRODUCTION

The concept of ride-sharing has becoming more and more popular nowadays. Early real-time ride-sharing projects began in the 1990s, but they faced obstacles such as the need to develop a user network and a convenient means of communication. However, with the rapid development of technology and the large-scale adoption of smart phones, the ride-sharing service provided by companies like Uber and Lyft has brought us timely and convenient transportation alternatives in our daily life and it has become an important part of the MoD (mobility-on-demand) systems.

The enormous potential of ride-sharing for positive societal impacts has been presented with respect to pollution, energy consumption and congestion (Ferguson, 1997; Kelly, 2007; Morency, 2007; Chan and Shaheen, 2012). The congestion problem is now costing the United States 121 billion per year (Schrank et al., 2012), which includes 5.5 billion hours of time lost to sitting in traffic and an extra 2.9 billion gallons of fuel burned. A recent study in New York City showed that up to 80 percent of the taxi trips in Manhattan could be shared by two riders with an increase in the travel time of a few minutes (Santi et al., 2014). Trips happen at peak hour time, when traffic jams cause cars to pollute an 80 percent more (Levofsky and Greenberg, 2001), additional benefits for the urban environment and climate change mitigation are expected by a reduction in the number of cars riding daily by the cities with a single occupant, and their related CO_2 and NO_x emmisions.

By effectively using new communication capabilities like mobile technology and GPS, there are several attempts to enable dynamic or real-time ride-sharing systems (Amey et al., 2011; Ghoseiri et al., 2011; Agatz, 2013). Real-time ride-sharing refers to a system which supports an automatic ride-matching process between participants on very short notice.Recently, a general mathematical model for high-capacity real-time ride-sharing has been developed recently to illustrate the potential of ride-sharing and it also provides us a fleet assignment method with anytime optimality that is fast enough to provide the users with the experience of real-time booking and service (Alonso-Mora et al., 2017). The algorithm of model will be applied in the simulation model in this study.

Since ride-pooling has been commercialized and has been gradually accepted as an important components of urban transportation, the pricing problem has naturally become a focus because it is the vital factor that can influence the market.

The ride-pooling problem is based on the network simulation, so lots of literature just focus on characterizing open and closed queueing-network models (Adelman, 2007; George et al., 2012; Kelly, 2011) which provide us a way to formulate the pricing problem in a specific network. Other literature propose the factors that can influence the profit of the ride pooling system. The work of Bimpikis has shown the profits of the system and the surplus of the consumer are monotonic with the balancedness of the demand pattern (Bimpikis et al., 2016). A bi-level programming framework is used to study the impacts of surge pricing in (Zha et al., 2017). Ozkan and Ward investigate the driver allocation policies without using surge pricing (Ozkan and Ward, 2017) while similar assignment problems are mainly based on queuing with congestion and network optimization (Banerjee et al., 2015).

Some literature directly provide us a dynamic pricing strategy that can be applied in the ride pooling system. The fleet assignment problem can be considered based on parallel auctions (Kleiner et al., 2011) while an approximation framework (Banerjee et al., 2016) can be also applied to determine the price of the assignment. An continuous approach (Luo and Saigal, 2017) and a queueing theoretic approach (Waserhole and Jost, 2016) can be both referenced to find out an optimal pricing strategy in the ride pooling system.

Other literature demonstrate the social aspects of dynamic ride-sharing (Sarriera et al., 2017), the sharing economy of companies like Uber (Cramer and Krueger, 2016) and also the state-of-art and future directions of ride-sharing (Furuhata et al., 2013). Since self-driving is becoming more and more popular these days some of the studies just focus on the autonomous MoD systems (Pavone et al., 2012; Spieser et al., 2016).

However, none of these studies have considered implementing a pricing strategy to maximize the total profit of the MoD system by assigning an optimal discount for each specific request based on the model which estimates the acceptance rate of the passenger once given a price of the ride. In order to calculate the profit of the ride-sharing service, it is important to build proper models to estimate the trip cost and expected revenue of a request precisely and that is the major challenge when addressing the problem.

CHAPTER 2

SIMULATION MODELS

2.1 The components of the simulation model

This section will illustrate the basic components included in the simulation model applied in this study. The basic components of the simulation model are the network, the demands and the vehicles.

The simulation starts when the demands are feed into the model in the first time interval and ends while all the demands have been either satisfied or ignored.

The network in this study is based on the real traffic network of Manhattan in New York City produced by the OSMNX package in python. After simply modifying the original network produced by OSMNX, the traffic network that is implemented in the simulation model is a strongly connected network with 4376 nodes and edges between them. Since we have a strongly connected network, we can get the shortest path from one node to another using Dijkstra Algorithm so that we can calculate the travel time and distance between each node in the network. However, it is too complicated and almost impossible to simulate the real time traffic condition for lack of the trail data of vehicles in the network, so congestion and traffic lights are both ignored in the simulation model of this study. The vehicle are just suppose to move with a constant speed in the network while travelling.

The demands of this study are extracted from the TLC Trip Data offered by NYC TaxiLimousine Commission. Denote all the node in the network mentioned above to be a hub where a specific demand can be picked up and dropped off in the simulation model. The pick up and drop off locations in the data are clustered to the nearest hub in the network and thus the corresponding hubs are regarded as the pick up or drop off place in the simulation model. In each time interval of the simulation process, the demands with request time in the time interval are feed into the simulation model and the status are updated after each time interval.

The vehicles in the simulation model are randomly distributed in the hubs of the network at the start of the simulation and all of them are initialized as an idle vehicle. The simulation model runs with the assumption that the network is a closed system which means no vehicle is going to leave or enter the system during the process. The position of the vehicles are updated in each time interval.

The basic logic of the simulation model implemented in this study will be mentioned in the following sections.

2.2 The pricing strategy

As is known to all, the profit is of vital importance for taxi companies or technological platforms like Uber or Lyft. So it is rational to think about maximizing the total profit while offering ride for passengers. This section will explain the pricing strategy in this study in detail.

First of all, define $P_{v,r}$ to be the price the passenger pay for the ride if a vehicle v takes his request r, define $C_{v,r}$ to be the estimate cost that is going to be paid to the driver in vehicle v which takes the request r, then the profit $\pi_{v,r}$ for request r assigned to vehicle v can be written as:

$$\pi_{v,r} = P_{v,r} - C_{v,r} \tag{2.1}$$

Since the price the passenger is charged and the estimate cost the driver is going to be paid can be different under different circumstance, the expression will be different in different simulation models in this study. While ordering a ride using applications like Uber or Lyft, passengers can even determine whether to accept the price after requesting, which makes the problem more complicated.

2.2.1 Pricing strategy for non-shared ride

The price $P_{v,r}$ and the cost $C_{v,r}$ for a non-shared ride can be fixed once the vehicle assigned to the demand is determined because the travel time and distance of the trip are both determined. However, in the MoD systems provided by Uber or Lyft, the price they charge the passengers can be higher because of the bad weather or insufficiency of available vehicles. In this study, we just assume the price for non-shared ride is always fixed regardless of the influence of weather or supply. The cost of a ride in this study is completely based on the travel distance and travel time of the vehicle and it is always fixed. Both expressions of the price and cost will be defined explicitly in Section 2.3.

The probability of the passenger accept the price $P_{v,r}$ can be defined with a function $F(P_{v,r})$ since giving a proper price is the purpose of this study. A discrete choice model is used to interpret the relationship between the price and probability of acceptance in this study. The model will be introduced in chapter 3. The other influences like time, weather or trip distance are actually constant for a specific request in the model. So the expected profit expression can be written as:

$$\pi_{v,r} = F(P_{v,r})(P_{v,r} - C_{v,r}) \tag{2.2}$$

Since the ride won't be shared with other passengers in this system, which means the vehicle assigned to the demand goes directly from where it is to pick up and the drop off the passenger along the optimal path in the network, there's no reason to give a discount to the trip. And we have previously assumed passengers won't be charged a higher price for the weather or supply factors. If the passengers are rational enough, their decisions won't be influenced by the price because it is always fixed once the pick up and drop off location are both determined. So all the passengers in the simulation model are considered to accept the price once the request is assigned. The simulation of the system with non-shared rides is actually a reference for the simulation of ride sharing model. The pricing problem of the ride sharing service is what we do really care.

2.2.2 Pricing strategy for ride pooling

The pricing strategy in the ride pooling simulation model is much more complicated because the cost is no longer fixed once a vehicle is assigned to a request. Vehicles with passengers may be assigned to other requests as long as they have enough space, consequently, vehicles will make detours to pickup other passengers during the trip so that the route of the vehicles may change in each time interval and there can be more than one passenger in a vehicle. However, the drivers are still going to be paid according to the mileage and time they drive, once there are more than one passenger in a vehicle, the cost for the shared distance and time can be split by the passengers in the vehicle so that the cost for shared request can be less than it originally takes.

In order to get the expression of the estimate cost of each request in the simulation model. Define the estimate cost of request r assigned to vehicle v to be $\hat{C}_{v,r}$. Consequently, whether this request can be shared with others need to be estimate firstly, then the estimate cost $\hat{C}_{v,r}$ can be calculated based on the estimate shared distance. The model to estimate the share probability of a request and the way to estimate the shared distance will be mentioned in Chapter 4.

Since the vehicle with passengers in it may take detours to pickup someone else during the trip, the trip of the passengers in it may be delayed because of the detour it takes. Thus the price $P_{v,r}$ charged from the passenger should be relatively lower than it used to be as a non-shared request. If the passengers in this system are rational enough, they are definitely going to expect a lower price to compensate the time they lost because of ride pooling.

Here we just define the variable $d_{v,r}$ to be the ratio of the ride-sharing price and the original price of the request as a non-shared request, which also means a $(1 - d_{v,r})$ discount because of ride pooling.

So the function $F(P_{v,r})$ which interprets the relationship between the acceptance rate and the price is no longer a constant and it can be considered as a function $F(d_{v,r})$ that is related to $d_{v,r}$. Then the expression of the expected profit of request r can be written as:

$$\pi_{v,r} = F(d_{v,r})(P_{v,r}d_{v,r} - \hat{C}_{v,r}) \tag{2.3}$$

Define Pr_r to be the estimated sharing rate of request r, which can be obtained from the decision tree model in Chapter 4, D_{ratio} denotes the ratio of the actual travel distance because of ride pooling and the non-sharing travel distance while S_{ratio} denotes the ratio of the shared travel distance of the request and the actual travel distance because of ride pooling. The the expression of $\hat{C}_{v,r}$ can be written as:

$$\hat{C}_{v,r} = C_{v,r}(Pr_r D_{ratio}(0.5S_{ratio} + 1 - S_{ratio}) + (1 - Pr_r)) \tag{2.4}$$

2.3 The simulation model of non-shared ride system

The basic logic of the simulation model regardless of ride sharing will be illustrated in this section. As is mentioned above, the taxi service and Uber service without ride sharing share the same pricing strategy and under this circumstance, once a taxi or a Uber vehicle is assigned to a request, it will move directly to pickup and drop off the passenger following the optimal route calculated by the Dijkstra Algorithm. Consequently, the price, cost, travel time can be precisely estimated once a request is assigned to a vehicle since the traffic condition is ignored in this model and the vehicles in this model always travel with a constant speed.

The first step of simulation is always initializing the model by defining a fixed number of vehicles randomly distributed in the network mentioned in 1.1. The other inputs like maximum waiting time of the passengers, the start time of the simulation should also be defined before the first interval of the simulation.

Then for each time interval, there are 4 basic steps explicitly explained in the following subsections:

2.3.1 Feed requests

Feed the requests that appear in this time interval to the model, the requests that are processed in the time interval are these newly arrived requests together with the unassigned request remained in the last time interval.

2.3.2 Assignment ILP

Solve the linear programming to get the optimal assignment between available vehicles and the requests. In this model, the purpose of the assignment optimization is to maximize the total profit of the system.

Define the set V_i includes all the idle vehicles that are available in this time interval, the set R_u includes the unassigned requests need to be assigned.

The variable $x_{v,r}$ denotes the assignment of the vehicle to the request. If the vehicle v is assigned to request r, then $x_{v,r} = 1$, otherwise 0. The expression of the linear programming can be written as:

$$\max_{x} \sum_{v \in V_i} \sum_{r \in R_u} x_{v,r} F(P_{v,r})(P_{v,r} - C_{v,r}) \tag{2.5}$$

Subject to:

$$\sum_{v \in V_i} x_{v,r} \leq 1 \quad \forall v \in V_i \tag{2.6}$$

$$\sum_{r \in R_u} x_{v,r} \leq 1 \quad \forall r \in R_u \tag{2.7}$$

The constraints means each idle vehicle can take at most one request and each request can be assigned only once.

$P_{v,r}$ is the price the passenger is going to be charged for the ride, in this study the price for the passengers includes three parts, the base fare, the distance fare and the time fare, since we have the assumption that the vehicles are moving with a constant speed in the network, the time fare and the distance fare can be combined. For each request, there is a minimum fare, if the final price is less than the minimum fare, then the price should be the minimum fare, otherwise it is the result of adding the three parts together.

Define P_u to be the unit price of distance after taking travel time into account. Similarly, P_{min} denotes the minimum price of the service, P_{base} denotes the base fare for each request, $D_{v,r}$ denotes the optimal distance vehicle v is going to drive from the place it is assigned to request r to the destination of the request and $t_{v,r}$ to be time it takes. Then the expression of $P_{v,r}$ can be written as:

$$P_{v,r} = max\{(P_b + P_u D_{v,r}), P_{min}\} \tag{2.8}$$

The cost $C_{v,r}$ here means how much the driver is going to be paid for the ride, it is rational to pay them according to the miles they drive and the time it takes.

Define p_d here to be the unit price per kilometer the driver is going to be paid, then p_t denotes the unit price per second the driver is going to be paid.

$$C_{v,r} = D_{v,r} p_d + t_{v,r} p_t \tag{2.9}$$

2.3.3 Rebalance LP

Once finishing assigning the requests to idle vehicles, there may be some requests or idle vehicles remaining to be unassigned. Under the assumptions that ignored requests may request again and there may be more requests appearing in the same area where there exist unsatisfied request.

The variable $y_{v,r}$ is actually a binary variable which indicates idle vehicle $v \in V_i$ is assigned to request $r \in R_u$ when $y_{v,r} = 1$.

Define $\tau_{v,r}$ to be the shortest travel time for $v \in V_i$ to reach the pickup place for $r \in R_u$, it can be easily calculated since the network has been well constructed. We are going to rebalance the idle vehicles based on the following rules:

$$\min_{v,r} \sum_{v \in V_i} \sum_{r \in R_u} \tau_{v,r} y_{v,r} \tag{2.10}$$

Subject to:

$$\sum_{v \in V_i} \sum_{r \in R_u} y_{v,r} = min(|V_i|, |R_u|) \quad \forall v \in V_i, \forall r \in R_u \tag{2.11}$$

$$0 \leq y_{v,r} \leq 1 \tag{2.12}$$

The purpose of the rebalancing process is to rebalance the idle vehicles in the network to the place where there are not enough available vehicles. By solving the linear programming problem listed above, all the idle vehicles can be assigned to pick up a unsatisfied request until there are no idle vehicles or unsatisfied remaining with a minimum total travel time. It is actually an integer programming (IP) problem but the solution of the linear programming (LP) can be either 1 or 0 to minimize the total travel time so it is not surprising to see solving the IP and LP are the same here.

2.3.4 Update the status of vehicles and requests

Update the position of the vehicles and the status of the requests in the simulation model then start the next time interval. Once all the requests that are not ignored in the model reach their destination, the simulation comes to the end.

2.4 The simulation model of ride pooling

This section demonstrates the basic logic of the ride sharing simulation model applied in this study. The network used in the model is no different from the previous model but the logic can be much more complicated since in the ride sharing model, vehicle are no longer assigned to a single request but a trip instead. Several requests can be included in a trip so that the sequence of picking up and dropping off those passengers of the trip should be taken into consideration when assigning a vehicle to the trip. All the available trips should be built before the assignment optimization in each time interval according to the maximum waiting time and maximum delay because of ride pooling.

In (Alonso-Mora et al., 2017), the way to build an on-demand high-capacity ride sharing model with anytime optimality has been explicitly illustrated. The algorithm of constructing the available trips in this study will be explained later in this section.

Similarly, before running the simulation model, some constants like the number of vehicles in the network, the maximum waiting time after requesting a vehicle, the maximum delay time because of ride sharing and the start time of the simulation should be defined. The the simulation is initialized with all the vehicles remaining idle randomly distributed in the network. Then there are 5 steps for each time interval:

2.4.1 Feed requests to the model

The first step is no different from that in the simulation model without ride sharing. Just feed the requests that appear in this time interval to the model, the requests that are processed in the time interval are these newly arrived requests together with the unassigned request remained in the last time interval. However, all the vehicles should be taken into consideration in each time interval because even if it is fully occupied it can be assigned to some trip as long as it is going to drop some passengers off firstly.

2.4.2 Compute RR,RV,RTV graph

The model in (Alonso-Mora et al., 2017) present a framework for solving the real-time ride-pooling problem with arbitrary numbers of passengers and trips, anytime optimal rider allocation and routing dependent on the fleet location, and online rerouting and assignment of riders to existing trips. The capacity of the vehicle in the network is 10, and it makes the simulation super complicated. However, an ordinary vehicle can usually take a ride pooling task with 2 requests in real life. In this study, we just limit the capacity of the vehicle to be 2 and assume there is only one passenger for each request to make the simulation model more efficient.

Define V_a to be the set of available vehicles and R_u to be the set of unassigned requests in a specific time interval.

Now that we have the set of requests R_u and the set of vehicles V_a after finishing step1, the construction of trips starts with finding the possible combination of two requests, that is connecting RR graph.

Two requests r_1 and r_2 can be connected if an empty vehicle starting at the origin of one of them could pick up and drop off both requests while satisfying the following constraints:

For each request r, the pick up time $t_r^p \leq t_r^r + t_{max}^w$, the drop off time $t_r^d \leq t_r^{opt} + t_{max}^d$. For each vehicle the number of passengers $n_v^{pass} \leq 2$

Define the pickup time of request r to be t_r^p, t_r^d to be the drop off time of request r, t_r^{opt} to be the ideal drop off time of the request r, $t_r^{opt} = t_r^r + t_r^{Dij}$, t_r^{Dij} means the shortest path travel time from the pick up location to the destination of the request r. Then t_{max}^w denotes the maximum waiting time and t_{max}^d to be the maximum delay defined at the start of the simulation.

Once two requests r_1, r_2 can be connected, denote the edge $e(r_1, r_2)$ to represent the combination. Similarly, a request r and and a vehicle v are connected if the request can be served with the vehicle while satisfying all the three constraints mentioned above. The edge is denoted by $e(r, v)$ and that's the way we construct the RV graph. Since we've found a vehicle v to server r, we can define the travel to be $travel(v, r)$.

Once we've got all the edges in RR graph and RV graph, we can start to construct the RTV-graph to find feasible trips. A trip $T = [r_1......r_n]$ is a set of n requests to be assigned to a single vehicle, it is feasible if all the requests in it can be picked up and dropped off by some vehicle while satisfying all the constraints as mentioned above.

The RTV-graph contains two types of edges: (i) edges $e(r, T)$ between a request r and a trip T that contains request r ($e(r, T) \Leftrightarrow r \in T$). (ii) edges e(T,v), between a trip T and a vehicle v that can execute the trip ($e(T, v) \Leftrightarrow travel(v, T)$ is feasible). The total travel distance and the delay of serving all the requests in the trip T are associated to each edge $e(T, v)$. The algorithm to compute the feasible trips and edges proceeds incrementally in trip size for each vehicle, starting from the RV edges in the RV-graph. Since the capacity of the vehicles in this study is 2, we just need to consider the trips with 2 or less requests.

After running the algorithm in the simulation model, all the feasible trips in the time interval should be found. The algorithm goes as below:

```
RTV-Algorithm
initialization: Trips = ∅;
for each vehicle v ∈ V_a do
    Trips_k = ∅   ∀k ∈ 1, 2;
    Add trips for size one;
    for e(r, v) edge of RV-graph do
        Trips_1 ← T = [r];
        Add e(r, T) and e(T, v);
    end
    Add trips for size two;
    for all [r_1], [r_2] ∈ Trips_1 and e(r_1, r_2) ∈ RR-graph do
        if travel(v, [r_1, r_2]) is valid then
            Trips_2 ← T = [r_1, r_2];
            Add e(r_i, T) and e(T, v);
        end
    end
end
```

2.4.3 Assignment ILP

Since we've constructed all the trips that are available for this time interval, solve the integer programming problem to find the optimal assignment between the vehicles and the trips to maximize the total profit.

The variable $x_{v,t}$ denotes the assignment in the ILP, if $x_{v,t} = 1$, then vehicle v is assigned to trip r, otherwise v is not assigned to r.

Define V_a to be the set of available vehicles and R_t to be the set of unassigned requests in a specific time interval. Then S_t denotes the set of trips constructed in the RTV graph while v_t denotes the set of possible trips for vehicle v. And similarly define r_t to be set of possible trips for request. Then $P_{v,t}$ denotes the profit of assigning vehicle v to trip t. Then the expression of the assignment ILP goes as below:

$$\max_x \sum_{v \in V_a} \sum_{t \in S_t} P_{v,t} x_{v,t} \tag{2.13}$$

Subject to:

$$\sum_{t \in v_t} x_{v,t} \leq 1 \quad \forall v \in V_a \tag{2.14}$$

$$\sum_{t \in r_t} x_{v,t} \leq 1 \quad \forall r \in R_u \tag{2.15}$$

Once the simulation is used to get the data to train the decision tree model which will be mentioned in the next chapter, the profit is just what the passengers in the trip minus the cost of the trip which can be calculated in Section 2.4.2. Thus the expression of $P_{v,t}$ under this circumstance should be:

$$P_{v,t} = \sum_{r \in t} (P_{v,r} - C_{v,t}) \tag{2.16}$$

Once the simulation is used to simulate the ride sharing process to get the total expected profit after implementing machine learning model that will be explained in the next chapter. The expression can be written as:

$$P_{v,t} = \sum_{r \in t} F(d_{v,r})(P_{v,r} d_{v,r} - \hat{C}_{v,r}) \tag{2.17}$$

The price $P_{v,r}$ is still the price of the request which has been defined in expression 2.7 in last section. But the cost $\hat{C}_{v,r}$ here is the expected cost for each request.

$$\hat{C}_{v,r} = C_{v,r}[Pr_v D_{ratio}(0.5S_{ratio} + 1 - S_{ratio}) + (1 - Pr_v)] \tag{2.18}$$

The way to calculate $\hat{C}_{v,r}$ will be explained explicitly in the next chapter.

2.4.4 Rebalance LP

Once finishing assigning the requests to idle vehicles, there may be some requests or idle vehicles remaining to be unassigned. Under the assumptions that ignored requests may request again and there may be more requests appearing in the same area where there exist unsatisfied request. We are going to rebalance the idle vehicles based on the following rules:

$$\min_{v,r} \sum_{v \in V_i} \sum_{r \in R_u} \tau_{v,r} y_{v,r} \tag{2.19}$$

Subject to:

$$\sum_{v \in V_i} \sum_{r \in R_u} y_{v,r} = min(|V_i|, |R_u|) \quad \forall v \in V_i, \forall r \in R_u \tag{2.20}$$

$$0 \leq y_{v,r} \leq 1 \tag{2.21}$$

The definition of the variables are all the same with the rebalancing process in the previous section.

2.4.5 Update the status of vehicles and requests

Update the position of the vehicles and the status of the requests in the simulation model then start the next time interval. Once all the requests that are not ignored in the model reach their destination, the simulation comes to the end.

CHAPTER 3

ML MODELS

3.1 The model to estimate the shared rate and distance

This section demonstrates the decision tree model used for estimating whether a given request can be shared with others along with the way to estimate the shared distance in this study.

The purpose of estimating whether a request can be shared and its shared distance is to get an estimate cost for each request once it appears in the simulation model so that the price it is charged can be formulated more rationally.

3.1.1 Decision Tree model applied to estimate the shared rate

Decision Trees (DTs) are a non-parametric supervised learning method used for classification and regression. The goal is to create a model that predicts the value of a target variable by learning simple decision rules inferred from the data features.

Given training vectors $x_i \in R^n$ and a label vector $y \in R^l$, a decision tree recursively partitions the space such that the samples with the same labels are grouped together.

In the decision tree model applied in this study, the features included in the training vectors are:

1. The request time in a day
2. The pickup longitude of the request
3. The pickup latitude of the request

4. The dropoff longitude of the request
5. The dropoff latitude of the request
6. The shortest path distance from the pickup location to the destination

All of these features are known once the request is added to the waiting list of the simulation model so that the shared rate can be estimated at the very beginning.

The labels of the model are either 0 which indicates the request can not be shared with others or 1 which indicates it can be shared.

The basic logic of the model is to split the data into two separate parts with some criterion at each node and minimize the impurity from the root of the tree. Let the data at node m be represented by Q, for each candidate split $\theta = (j, p_m)$ consisting of a feature j and a pivot of the feature p_m, partition the data into the subsets $Q_L(\theta)$ and $Q_R(\theta)$,

$$Q_L(\theta) = (x, y)|x_j <= p_m, Q_R(\theta) = (x, y)|x_j > p_m \tag{3.1}$$

The impurity at node m is computed using the impurity function $H()$, the choice of the function depends on the task being solved, since the classification in this model is a binary classification, a common measure of this classification Gini is applied in this model.

Define k to be the proportion of k possible observations at node m, (0, 1) for this model. Then define X_m to be training data at node m, thus N_{X_m} denotes the number of observations in set X_m, R_{X_m} is its region. The expression of the impurity function can be written as:

$$H(X_m) = \sum_k p_{mk}(1 - p_{mk}) \tag{3.2}$$

$$p_{mk} = 1/N_{X_m} \sum_{x_i \in R_{X_m}} I(y_i = k) \tag{3.3}$$

The purpose of the Gini classifier in the decision tree model is to find a specific split θ that minimizes the summation of the impurity function $H(Q_L)$ and $H(Q_R)$ and recurse for the subsets $Q_R(\theta^*)$ and $Q_L(\theta^*)$ until the maximum depth of tree is reached or $N_m < min_{samples}$. The expression can be written as:

$$G(Q, \theta) = \frac{n_L}{N_m} H(Q_L) + \frac{n_R}{N_m} H(Q_R) \tag{3.4}$$

$$\theta^* = argmin_{\theta} G(Q, \theta) \tag{3.5}$$

Once the model is trained, the probability of getting both label can be predicted after feeding the features mentioned above of a specific request.

3.1.2 The prediction of the shared distance of a request

The total travelling distance of a shared request is no longer the shortest path distance from the pickup location to its destination because the vehicle may make detours to satisfy other request. And for each request, the passenger may share the vehicle during part of the trip while occupy the vehicle alone for the rest of the trip. It is really important to estimate the shared distance of a trip because once a vehicle is shared, the driver is still paid according the time and distance but all the passengers in this vehicle are paying for the trip. So it is possible for technological platforms like Uber or Lyft to make more profit by charging the passengers with lower price for ride sharing.

However, it is really hard to give a precise estimation of the total travelling distance of each request and the percentage of shared distance of each request since it is no longer a binary classification as mentioned above.

For each request r, define D_{tt} to be the actual travel distance of the trip and D_{sp} to be the shortest path travel distance of the trip. Then the ratio D_{ratio} between them can be written as:

$$D_{ratio} = \frac{D_{tt}}{D_{sp}} \tag{3.6}$$

Define D_{sh} to be the shared distance of the trip, then the ratio S_{ratio} between the shared distance of a trip and its total travel distance can be expressed as:

$$S_{ratio} = \frac{D_{sh}}{D_{tt}} \tag{3.7}$$

After running the simulation model by assuming all the passengers will accept the ride if the request is assigned, D_{share} and D_{total} can be obtained from the results so that D_{ratio} and S_{ratio} can be calculated. It is not hard to find that these two ratios varies obviously according to the total distance of the trip. The stats are shown below:

Distance(km)	≤ 2	2-4	4-6	6-8	8-10	10-12	12-14	14-16	≥ 16
D_{ratio}	1.07	1.16	1.14	1.12	1.11	1.10	1.09	1.08	1.08
S_{ratio}	0.79	0.70	0.68	0.68	0.66	0.63	0.62	0.60	0.60

The stats are obtained from the result of feeding over 2,000,000 requests into the simulation model. All of the ratios listed above are the average numbers.

Since it is really hard to give a precise estimation of both distance, once finishing predicting whether a request can be shared or not, D_{ratio} and S_{ratio} are used to predict the total travel distance and shared distance for this request.

3.2 The model to estimate the acceptance rate

This section demonstrates the discrete choice model used for estimating the acceptance rate according to the features of a specific request.

The purpose of applying such a model to estimate the acceptance rate is because once given an price of after requesting a vehicle, the passenger may not be willing to accept it. There is no doubt that the price itself can be the most important factor that influences the judgment of the passengers. After all it takes them more time to reach their destination for choosing ride sharing, it is rational to get a lower price because this. Other factors like the time, weather or even the pickup location may influence their determination, for example, when the weather is bad or when it is rush hour, passengers at a remote place or a place with lots of other passengers waiting for available vehicles may accept a higher while the price should be much more lower to attract the passengers with lots of alternatives other than ride sharing. The more features taken into consideration, the more precise the model may be, however it is hard to get access to the commercial data of Uber or Lyft since they are highly classified. Several features that are relatively easy to get are used in this model:

1. The request time in a day
2. The pick up location of the request
3. The drop off location of the request
4. The estimate total travel distance
5. The ratio of the ride-sharing price and the original price of the request as a non-shared request $d_{v,r}$

Continuous features like the estimate total travel distance, $d_{v,r}$ are normalized before feeding into the model. The estimate distance for request r can be estimate as mentioned in Section 3.1.2.

Categorical features like the request time, the pick up location and the drop off location are divided into several types before feeding into the model.

For the feature request time, number of requests varies differently for different time of a day. And the distribution of peak hour in weekdays can be different from that at the weekend.

weekend				
Time	2-4	4-8	8-12	12-2
#Request	10k-15k	3k-7k	10k-15k	$\geq 20k$
Type	1	0	1	2

weekday						
Time	0-6	6-8	8-10	10-17	17-21	21-24
#Request	3k-7k	10k-15k	$\geq 20k$	10k-15k	$\geq 20k$	10k-15k
Type	0	1	2	1	2	1

For the feature pick up location, just sum up the number of requests leaving or reaching a hub for all the hubs in the network and mark the hub with number 0-5 representing the popularity of it.

Popularity: Low $\rightarrow$ *High*						
#Request	≤ 10	10-100	100-500	500-1k	1k-2.5k	$\geq 2.5k$
Type	0	1	2	3	4	5

For the feature drop off location, just do the same as mentioned above.

Popularity: Low $\rightarrow$ *High*						
#Request	≤ 10	10-100	100-500	500-1k	1k-2.5k	$\geq 2.5k$
Type	0	1	2	3	4	5

The label 0 in the model represents the passenger refuse to accept the price while 1 indicates the opposite. We are going to use the logistic regression to find the relationship between the acceptance rate and those feature mentioned above. All the features used in the model are assumed to be independent.

Define U_r to be the utility of the request r and β to be the parameters we want to find through the logistic regression. x_r denotes the matrix of the features of the request fed into the model. Then the expression of the utility function U_r

can be written as:

$$U_r = \beta x_r + \varepsilon \tag{3.8}$$

The unobserved term ε is assumed to have a logistic distribution so that the probability function of the acceptance rate in the logistic model is a sigmoid function:

$$Pr = \frac{1}{1 + e^{-U}} \tag{3.9}$$

Utility $U_r > 0 \Leftrightarrow Pr > 0.5$ which indicates the passenger is more likely to accept the ride while $U_r > 0$ means the opposite.

Once the binary choice model is trained, define $\beta_{d_{v,r}}$ to be the coefficient of $d_{v,r}$ calculated by the logistic regression. It must be negative because the higher the price is the less likely the passenger is going to accept.

Define c to be the sum of the attributes other than $d_{v,r}$ times their coefficients calculated by the logistic regression for a specific match of vehicle v to request r. These attributes are the features mentioned above, they are all determined once the request is assigned to a vehicle.

After feeding the features mentioned above to the model, the probability of acceptance can be estimated. Since it is a binary choice logit model, the expression of the probability that a passenger accept the price if his request r is assigned to a vehicle v can be written as:

$$Pr(v, r) = F(d_{v,r}) = \frac{1}{1 + e^{-\beta s_n}} = \frac{1}{1 + e^{-\beta_{d_{v,r}} d_{v,r} + c}} \tag{3.10}$$

Then the expected profit of a specific match can be written as:

$$\pi_{v,r} = F(d_{v,r})(P_{v,r} d_{v,r} - \hat{C}_{v,r}) = \frac{P_{v,r} d_{v,r} - \hat{C}_{v,r}}{1 + e^{-\beta_{d_{v,r}} d_{v,r} + c}} \tag{3.11}$$

The estimate cost of the trip is also a constant that can be calculated based on the model mentioned above once the match of vehicle and request is determined. For request r assigned to vehicle v, the expression of the estimate cost mentioned in Section 2.2 can be calculated now.

$$\hat{C}_{v,r} = C_{v,r}(Pr_r D_{ratio}(0.5S_{ratio} + 1 - S_{ratio}) + (1 - Pr_r)) \tag{3.12}$$

$Pr(v)$ is the estimated sharing rate obtained from the decision tree model for request r, D_{ratio} and S_{ratio} are both defined in Section 2.1.2.

The goal of the assignment optimization in each time interval is to find an optimal assignment that maximize the total expected profit. Since for each v, r, the expected profit is now a function of $d_{v,r}$, the optimal discount for match v, r can be determined if there always exist a maximum value of $\pi_{v,r}$. This can be proved and the proof is attached in the appendix.

CHAPTER 4

DATA ANALYSIS

4.1 The comparison between two simulation models

The data used to access the performance of the models in this study is one-week data (2016-01-03-00:00 - 2016-01-09-24:00) arbitrarily chosen from the publicly available dataset of taxi trips in Manhattan, New York City. Each day contains between 292,605 (Sunday) and 392,923 (Saturday) requests. We just extract the pick up and drop off location along with the request time from the data set and clustering all the pick up and drop off location to the nodes in the network mentioned in Section 2.1. The shortest path and travel time between each nodes in the network has been calculated and stored in advance.

The logic of both simulation models has been explicitly illustrated in Chapter 2. In this study we perform the simulation with both models by initializing all the vehicles in the network at midnight and assuming they are randomly distributed in the network. Requests are collected in each time interval, 30 s in the experiments. And then the simulation of both models just go step by step as they are mentioned in Section 2.3 and 2.4. All the passengers are assumed to accept the ride assigned to their request in the simulation.

The results after running these two models are shown in the tables in the appendix with Day 1-Day 7 corresponding to the simulation from 01-03-01-09. Table 1 and Table 2 illustrate the result of running the non-shared ride simulation model with 3000 and 5000 vehicles in the network, Table 3 illustrates the result of running the ride-sharing simulation model with 3000 vehicles in the network. The max waiting time are 300 seconds in the three experiments, the max delay time because of the ride sharing is 600 seconds in the ride-sharing model.

After analyzing the data it is not hard to find that 3000 vehicles with capacity 2 can serve nearly 95% of the requests in the network (Each request is assumed to occupy 1 capacity of the vehicle) and 94.6% of the served requests are shared with others. The average waiting time of all served requests is 130 seconds while the average time delay because of ride-sharing of all shared requests is 186 seconds.

However, 3000 vehicles can only serve 235,000 requests on average once the vehicles can not be shared and the average serve rate is merely 69% which is much less than the 95% served rate in the ride-sharing simulation model. The waiting time is about 130 seconds which is approximately the same as that in the ride-sharing simulation model. The total travel distance of the occupied vehicles when vehicles can not be shared is even more than that when ride-sharing is allowed, and it makes the total profit of the non-shared ride system much less than the system with ride-sharing, which is merely 44.4% on average when same scale of vehicles in the network.

When we initiate the non-shared simulation model with 5000 vehicles in the network, nearly 99% of the requests can be served this time, and the average waiting time is just 95 seconds. However, in order to serve approximately the same number of requests in the system, the total travel distance of occupied vehicles in the ride-sharing system is just 2/3 of that in the non-sharing ride system. And that makes the total profit of serving a same scale of requests in the non-shared ride system 30% percent less than the in the ride-sharing system. It's not hard to find the potential of ride-sharing since the results of the simulation models has shown the demand of the passengers can be satisfied with less vehicles and less total travel distance which means more profit.

However, we have assumed that all the passengers are going to accept the ride-sharing service if the price of the ride is the same as it is not shared. It is apparently not rational because ride-sharing takes the passengers longer travel time to arrive their destination, the passengers should expect a lower price since we assume all the passengers are rational enough in this study. If the price we offer is higher than their expectation, they may not accept the trip which means we can get 0 profit of the request. So it is important to estimate the acceptance rate of the passengers based on the price of ride-sharing and some other features of the request. Consequently, it is more rational to estimate the total profit of the ride-sharing system using the expect profit as is defined with expression 3.11.

We use the trip cost to calculate the cost of the requests in a trip when running the simulation mentioned above, the problem is the trip cost is no longer valid when estimate the cost $\hat{C}_{v,r}$ in expression 3.11 because we don't know whether it can be shared with others and the shared distance of it before it is completely finished. That's why we also need a method to estimate $\hat{C}_{v,r}$ as is mentioned in Chapter 3.

4.2 Training and testing the ML models

As is mentioned at the end of Chapter 3, we are interested in the expected profit $\pi_{v,r}$ of a specific match in the ride-sharing system and thus find a method to estimate it by developing a decision tree model to estimate $\hat{C}_{v,r}$ and a discrete choice model to estimate the relationship between the price ratio $d_{v,r}$ and the acceptance rate $F(d_{v,r})$.

4.2.1 Decision Tree Model

The data used for training the decision tree model is still the same data that is fed into the simulation model. After running the simulation model of the ride pooling system by assuming all the passengers are going to accept the ride once a request is assigned to a vehicle, whether a request is finally shared with others is known and thus we can get the label of it.

The basic logic of the Gini classifier used in the decision tree model and the features used for training this model have been illustrated explicitly in Section 3.1.1. While training this model, it is important to set a proper max depth for the decision tree in case of overfitting. Cross validation is used to examine how the model behave with different depths, just randomly split the data into 10 subsets with equal size, each time choose 9 subsets to be the training data while the rest should be the test data, examine the stats of the model. The averaged results are shown in the form below:

maxdepth	5	10	13	16	18	20	22
AUC	88.7%	89.1%	90.1%	91.3%	92.6%	92.7%	92.8%
TP	99.5%	99.3%	98.6%	98.7%	98.7%	98.8%	98.8%
FP	96.1%	91.8%	76.1%	67.0%	58.1%	56.5%	54.8%

AUC here denotes the area under ROC curve. TP denotes the true positive rate while FP denotes the false positive rate.

The AUC and true positive rate of the model is always very high because nearly 91% of the requests are found to shared with others in the simulation. So the AUC and true positive rate can not be valid criterion to estimate how the model behaves. It's better to see how the false positive rate change with the increasing of max depth of the model. It's not hard to find when the max depth of the tree is greater than 18, the AUC and false positive rate change little which means the model won't be much better to classify whether a request can be shared or not with the increasing max depth. Too many layers of the model may cause overfitting rather than make it better.

However, the false positive rate of the model is still pretty high which means over 50% of the unshared requests are not correctly estimated. It is because there are only 9% of the requests are not shared with others. If we just randomly estimate it, only 9% of the unshared requests will be correctly estimated, the decision tree model is far better than that to some extent.The ROC curve of the decision tree model when the max depth is 18 is shown below:

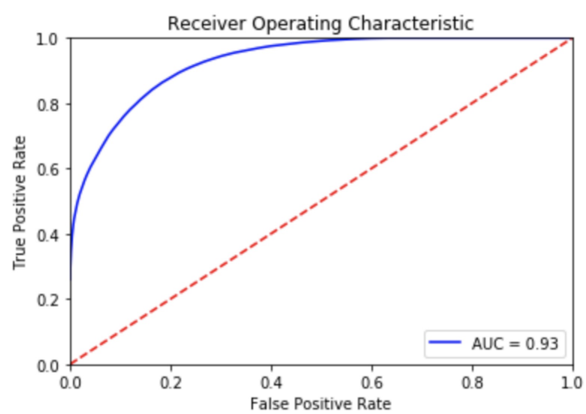

Figure 1:The ROC curve of the decision tree model when max depth = 18

After training the decision tree model, the probability of whether a request can be shared with others in the ride-sharing system can be estimated. S_{ratio} and D_{ratio} has been defined in Section 3.1.1, the stats are shown below:

Distance(km)	≤ 2	2-4	4-6	6-8	8-10	10-12	12-14	14-16	≥ 16
D_{ratio}	1.07	1.16	1.14	1.12	1.11	1.10	1.09	1.08	1.08
S_{ratio}	0.79	0.70	0.68	0.68	0.66	0.63	0.62	0.60	0.60

By now, we have all the stats to estimate cost of a request once the match of vehicle and the request is determined. For request r:

$$\hat{C}_{v,r} = C_{v,r}(Pr_r D_{ratio}(0.5S_{ratio} + 1 - S_{ratio}) + (1 - Pr_r)) \tag{4.1}$$

4.2.2 Discrete Choice Model

Since we do not know the response of passenger to the ride-sharing service offered to them in the NYC taxi data, we'll have to find some other data to train the discrete choice model in this study. The data used for training the discrete choice model is the ride-sharing service data in Beijing from DIDI, once the customer chooses the ride sharing service of DIDI, the features like request time, request position will be recorded by the company and then the company will offer the passenger a price for the passenger. The decision of the passenger is also recorded so that we know whether a passenger accept a ride with specific features.

The basic logic of the binary choice model and the features used for estimating the acceptance rate of the passenger has been explicitly explained in section 3.2. The only difference is how we define the peak hours and the popularity of pick up and drop off locations here. The definition of them in section 3.2 are based on the NYC taxi data. To make the definition of peak hours and the popularity of pick up and drop off locations consistent, we just extract 2,000,000 (same scale as the NYC taxi data we use) ride-sharing service data in Beijing to train the model. Since the ride-sharing service is not available from 24:00-7:00 at night, this period of time are thought to be the off-peak hours with type 0. The time type of training data is shown below:

The requests in the ride-sharing service data of Beijing are also clustered into

Time	0-7	7-10	10-17	17-21	21-24
#Request	0	$\geq 20k$	0-20k	$\geq 20k$	0-20k
Type	0	2	1	2	1

hubs in the map. Since the scale of the training data is the same as the NYC taxi data we use to define the popularity of the pick up and drop off locations in Manhattan. The definition of popularity type of the pick up and drop off locations remain the same. For the pick up location:

Popularity: Low $\rightarrow$ *High*						
#Request	≤ 10	10-100	100-500	500-1k	1k-2.5k	$\geq 2.5k$
Type	0	1	2	3	4	5

For the drop off location:

Popularity: Low $\rightarrow$ *High*						
#Request	≤ 10	10-100	100-500	500-1k	1k-2.5k	$\geq 2.5k$
Type	0	1	2	3	4	5

Then the data is split into 10 parts with same size, each time when we run the regression, 9 subsets of the data are used to train and the rest is used for testing. The regression result is the average of 10 attempts.

feature	Time	PuL	DoL	Distance	$\mathrm{d}_{v,r}$
coefficient	1.479	0.713	-0.376	-1.533	-18.22

The intercept is 12.669, define x_t to be the variable of feature request time, x_{pu} to be the variable of feature pick up location, x_{do} to be variable of the feature drop off location, x_d to be the variable of the feature travel distance, $x_{d_{v,r}}$ to be the variable of the feature price ratio. Thus utility of request r can be written as:

$$U_r = 1.479x_t + 0.713x_{pu} - 0.376x_{do} - 1.533x_d - 18.22x_{d_{v,r}} + 12.669 + \varepsilon \tag{4.2}$$

The AUC of the model is 87%, the plot is shown below:

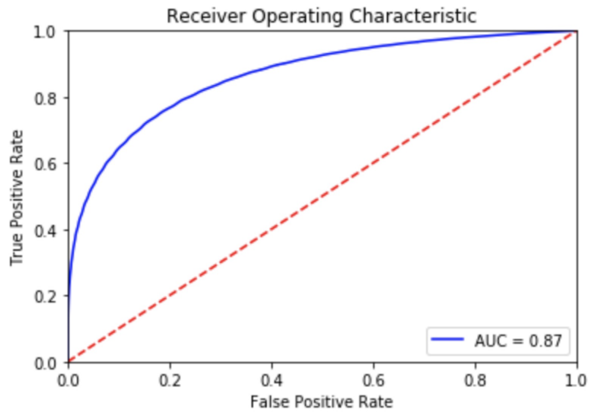

Figure 2: The ROC curve of the discrete choice model

The result means passengers are more willing to accept ride-sharing in rush hours and popular pick up place with a low price while long distance of the trip and popular drop off place may make them unwilling to accept it. The price is the most important factor that influence their decision.
Since the other features of a request is fixed, the relationship between acceptance rate and the price is:

$$Pr(v, r) = F(d_{v,r}) = \frac{1}{1 + e^{-\beta s_n}} = \frac{1}{1 + e^{-\beta_{d_{v,r}} d_{v,r} + c}} \tag{4.3}$$

By now, we have all the components of the expression of the expected profit illustrated so that we can find the optimal $d_{v,r}$ to maximize it in the simulation model as is proved in the appendix.

4.3 The expected profit of the ride sharing model

Now that we have the trained models to get the estimate cost $\hat{C}_{v,r}$ and the acceptance rate $Pr(d_{v,r})$, we just run the ride-sharing simulation model again by assuming the expected profit of request r when assigned to vehicle v to be:

$$Pr(d_{v,r}) = F(d_{v,r}) = \frac{1}{1 + e^{-\beta s_n}} = \frac{1}{1 + e^{-\beta_{d_{v,r}} d_{v,r} + c}} \tag{4.4}$$

The data we use here is still the one-week data (2016-01-03-00:00 - 2016-01-09-24:00) NYC taxi data which have been used for comparing the two simulation models as mentioned in Section 1 this chapter.

Table 4 and Table 5 in the appendix show the result of running the ride-sharing simulation model when the expected profit has been taken into consideration.

As is defined previously, $\hat{C}_{v,r}$ is the estimate cost of the request r when assigned to vehicle v, $\tilde{C}_{v,r}$ is the actual cost of it that can be obtained after finishing running the simulation.

In this study, we just run the ride-sharing simulation model again with 3,000 vehicles in it, the max waiting time and max delay time also remain the same. The only difference is the passengers in the systems are considered to have a acceptance rate of the ride assigned to them, so the profit of assigning request r to vehicle v is now expression 3.11.

We first run the ride-sharing simulation model by giving each request an optimal discount to maximize the total profit and then run it again by assuming all the requests are given an original price without any discount to see the difference of these two experiment. Since the basic logic does not change here when running these two experiments, some stats like the serve rate, share rate, average waiting time, average time delay, total travel distance and total rebalance distance do not change much.

As is shown in Table 4 and Table 6 in the appendix, the huge difference of the average acceptance rate in these two experiment illustrates the importance of finding a proper model to estimate the relationship between the price and the acceptance rate. When the optimal discount is given to each request the average acceptance rate can reach nearly 86%. When the passengers are charged with the original price price without any discount, the average acceptance rate is merely 9%. That leads to huge difference of the average profit and total profit between these two experiments. Although we give the passengers a 22% discount, the average profit of each ride can be 3.7 dollars on average which is nearly 6 times more than the average profit we get by charging them with the original price.

However, if we are rational enough we won't charge passengers a same price when they choose the ride-sharing service. So we just run the model again by assuming all the passengers are given an average 22% discount which is the average optimal discount given to the passengers to see the result.

As is shown in Table 5, once all the requests are given a 22% discount, the average acceptance rate is 68.2% and the average profit can reach 3$ which makes the total profit in this experiment 81% of that in the experiment where dynamic pricing strategy is applied. This result can illustrate the advantage of the pricing strategy, charging the passengers differently based on different situation.

It is not surprise to see total expected profit is not that much (58%) compared to the experiment in section by assuming all the passenger are going to accept the original price without any discount for the ride sharing service. However, even if the passenger are given a discount the total expected profit in the ride-sharing systems is still a little bit more than that in the non-shared ride system with same scale of vehicles in it. The commercial potential of the ride-sharing system is really fascinating because much more requests can be served with a relatively lower price by a certain number of vehicles if ride-sharing is allowed, and the total expected profit is even more than that in the non-sharing system.

CHAPTER 5
CONCLUSION

In this study, the enormous potential can be discovered after comparing the experiment results of the non-shared ride system and the ride-sharing system. Since it is rational to assume the passengers may expect a lower price of the ride-sharing system, it is also really important to find the relationship between the acceptance rate and the price to get an accurate estimation of the expected profit of each request so that we can find the optimal assignment to maximize the total profit of the ride-sharing system. The result shows that 3,000 vehicles of capacity 2 can serve as many request as 5,000 non-shared ride can do with merely 66% total travel distance. If we give the passengers an optimal discount for their ride-sharing service based on the machine learning model we trained in this study, the acceptance rate can be nearly 9 times more than that if the passengers are not given any discount, and this makes the total expected profit 6 times more if the optimal pricing strategy is applied.

The results show the vital importance of implementing proper models to estimate the trip cost and acceptance rate so that we can get an accurate estimation of the expected profit of a request. 90% of the requests in the ride-sharing system can be shared with others based on the decision tree model in this study. The parameters of the discrete choice model illustrates that passengers are more willing to accept shared ride in rush hours and popular pickup place with a low price while long distance of the trip may make them unwilling to accept it.

However, the data used to train these two models in the study is limited, some important feature like weather, traffic condition, vehicle supply and the customer features like age, gender, occupation may also influence the shared rate or acceptance rate. If we can get more complete commercial data of the ride pooling service, better prediction of the expected profit can be made, which indicates better pricing strategy can be found.

BIBLIOGRAPHY

Adelman, D. (2007). Price-directed control of a closed logistics queueing network. *Operations Research*, 55(6):1022–1038.

Agatz, N. (2013). Optimisation for dynamic ride-sharing. *RSM Discovery-Management Knowledge*, 15(3):13–15.

Alonso-Mora, J., Samaranayake, S., Wallar, A., Frazzoli, E., and Rus, D. (2017). On-demand high-capacity ride-sharing via dynamic trip-vehicle assignment. *Proceedings of the National Academy of Sciences*, 114(3):462–467.

Amey, A., Attanucci, J., and Mishalani, R. (2011). Real-time ridesharing: opportunities and challenges in using mobile phone technology to improve rideshare services. *Transportation Research Record: Journal of the Transportation Research Board*, (2217):103–110.

Banerjee, S., Freund, D., and Lykouris, T. (2016). Pricing and optimization in shared vehicle systems: An approximation framework. *arXiv preprint arXiv:1608.06819*.

Banerjee, S., Johari, R., and Riquelme, C. (2015). Pricing in ride-sharing platforms: A queueing-theoretic approach. In *Proceedings of the Sixteenth ACM Conference on Economics and Computation*, pages 639–639. ACM.

Bimpikis, K., Candogan, O., and Saban, D. (2016). Spatial pricing in ride-sharing networks.

Chan, N. D. and Shaheen, S. A. (2012). Ridesharing in north america: Past, present, and future. *Transport Reviews*, 32(1):93–112.

Cramer, J. and Krueger, A. B. (2016). Disruptive change in the taxi business: The case of uber. *American Economic Review*, 106(5):177–82.

Ferguson, E. (1997). The rise and fall of the american carpool: 1970–1990. *Transportation*, 24(4):349–376.

Furuhata, M., Dessouky, M., Ordóñez, F., Brunet, M.-E., Wang, X., and Koenig, S. (2013). Ridesharing: The state-of-the-art and future directions. *Transportation Research Part B: Methodological*, 57:28–46.

George, D. K., Xia, C. H., and Squillante, M. S. (2012). Exact-order asymptotic analysis for closed queueing networks. *Journal of Applied Probability*, 49(2):503–520.

Ghoseiri, K., Haghani, A. E., Hamedi, M., and Center, M. (2011). *Real-time rideshare matching problem*. Mid-Atlantic Universities Transportation Center Berkeley.

Kelly, F. P. (2011). *Reversibility and stochastic networks*. Cambridge University Press.

Kelly, K. L. (2007). Casual carpooling-enhanced. *Journal of Public Transportation*, 10(4):6.

Kleiner, A., Nebel, B., and Ziparo, V. A. (2011). A mechanism for dynamic ride sharing based on parallel auctions. In *IJCAI*, volume 11, pages 266–272.

Levofsky, A. and Greenberg, A. (2001). Organized dynamic ride sharing: The potential environmental benefits and the opportunity for advancing the concept. In *Transportation Research Board 2001 Annual Meeting*, pages 7–11.

Luo, Q. and Saigal, R. (2017). Dynamic pricing for on-demand ride-sharing: A continuous approach.

Morency, C. (2007). The ambivalence of ridesharing. *Transportation*, 34(2):239–253.

Ozkan, E. and Ward, A. (2017). Dynamic matching for real-time ridesharing.

Pavone, M., Smith, S. L., Frazzoli, E., and Rus, D. (2012). Robotic load balancing for mobility-on-demand systems. *The International Journal of Robotics Research*, 31(7):839–854.

Santi, P., Resta, G., Szell, M., Sobolevsky, S., Strogatz, S. H., and Ratti, C. (2014). Quantifying the benefits of vehicle pooling with shareability networks. *Proceedings of the National Academy of Sciences*, 111(37):13290–13294.

Sarriera, J. M., Álvarez, G. E., Blynn, K., Alesbury, A., Scully, T., and Zhao, J. (2017). To share or not to share: Investigating the social aspects of dynamic ridesharing. *Transportation Research Record*, 2605(1):109–117.

Schrank, D., Eisele, B., and Lomax, T. (2012). Ttis 2012 urban mobility report. *Texas A&M Transportation Institute. The Texas A&M University System*, 4.

Spieser, K., Samaranayake, S., Gruel, W., and Frazzoli, E. (2016). Shared-vehicle mobility-on-demand systems: a fleet operators guide to rebalancing empty vehicles. In *Transportation Research Board 95th Annual Meeting*, number 16-5987.

Waserhole, A. and Jost, V. (2016). Pricing in vehicle sharing systems: Optimization in queuing networks with product forms. *EURO Journal on Transportation and Logistics*, 5(3):293–320.

Zha, L., Yin, Y., and Du, Y. (2017). Surge pricing and labor supply in the ride-sourcing market. *Transportation Research Procedia*, 23:2–21.

APPENDIX A

PROOF OF THE EXISTENCE OF CRITICAL POINT TO MAXIMIZE THE EXPECTED PROFIT

The proof of the existence of $x_{v,r}$ that maximize $\pi_{v,r}$.

As is mentioned in 2.2, the expression of the expected profit of assigning a vehicle v to request r in the ride sharing simulation is:

$$\pi_{v,r} = F(d_{v,r})(P_{v,r}d_{v,r} - \hat{C}_{v,r}) = \frac{P_{v,r}d_{v,r} - \hat{C}_{v,r}}{1 + e^{-\beta_{d_{v,r}} * d_{v,r} + c}} \tag{A.1}$$

To make it easier to understand just let:

$$f(x) = \frac{P_{v,r}d_{v,r} - C_{v,r}}{1 + e^{-\beta_{d_{v,r}} * d_{v,r} + c}} = \frac{ax - b}{1 + e^{cx+d}} \tag{A.2}$$

Since the parameter $\beta_{d_{v,r}}$ is negative, $c = -\beta_{d_{v,r}}$, c must be positive. The price and the cost of the trip should also be positive, so both a and b must be positive. Since the purpose of the assignment optimization is to maximize the total profit in each time interval, if $P_{v,r}d_{v,r} - C_{v,r} \leq 0$, then this match will be ignored directly because it has no contribution to the object. So $ax - b$ must be positive, which means $\frac{b}{a} < x \leq 1$

The derivative of $f(x)$:

$$f^{'}(x) = \frac{a(1 + e^{cx+d}) - c(ax - b)e^{cx+d}}{(1 + e^{cx+d})^2} \tag{A.3}$$

Let $f^{'}(x) = 0$, thus:

$$a(1 + e^{cx+d}) - c(ax - b)e^{cx+d} = 0 \Leftrightarrow \frac{a}{c} + \left(\frac{a}{c} + b - ax\right)e^{cx+d} = 0 \tag{A.4}$$

Let:

$$k(x) = \left(\frac{a}{c} + b - ax\right)e^{cx+d} \tag{A.5}$$

$$k'(x) = c(b - ax)e^{cx+d} \tag{A.6}$$

Since $(b - ax)$ must be negative, The derivative $k'(x)$ of $k(x)$ must be negative, which indicates $k(x)$ is a decreasing function. It not hard to find when $x = \frac{b}{a}$, $f^{'}(x) > 0$ and when $x \to \infty$, $f^{'}(x) < 0$. Suppose when $x = x_c$, $f^{'}(x_c) = 0$, $f^{'}(x) > 0$ when $\frac{b}{a} < x_c$, $f^{'}(x) < 0$ when $x > x_c$ because $k(x)$ is always decreasing.

Thus $f(x)$ is increasing when $\frac{b}{a} < x_c$, and decreasing when $x > x_c$ which indicates $f(x_c)$ is the global maximum value of the function. x_c can be found by using the Nelder-Mead simplex algorithm or 'BFGS' method. If $x_c < 1$, then the optimal discount is just x_c otherwise it should be 1. Or the optimal discount can be found directly in the interval $\frac{b}{a} < x \leq 1$ by using Brents method.

Once the optimal discount that maximizes every single match in each time interval, it is not hard to find the optimal assignment since it is just a linear programming problem.

APPENDIX B

NOTATIONS

r: request

v: vehicle

t: trip

$P_{v,r}$: Price of request r served by vehicle v

$C_{v,r}$: Cost of request r served by vehicle v

$\hat{C}_{v,r}$: Estimate cost of request r served by vehicle v

$\pi_{v,r}$: Expected profit of request r served by vehicle v

$d_{v,r}$: Price ratio of request r served by vehicle v

$F(d_{v,r})$ The acceptance rate when price ratio is $d_{v,r}$

Pr_r: Probability request r can be shared with others

S_{ratio}: Shared distance ratio

D_{ratio}: Actual total travel distance ratio

D_{tt}: The actual travel distance

D_{sp}: The shortest path travel distance

D_{sh}: The shared travel distance

V_i: The set of idle vehicles

R_u: The set of unassigned requests

$x_{v,r}$: The assignment variable when optimize the total profit

$y_{v,r}$: The assignment variable when applying rebalancing

P_u: The unit price base on travel distance

P_{min}: The minimum price of calling a ride

P_{base}: The base fare for each ride

$D_{v,r}$: The shortest path travel distance of request r served by vehicle v

$\tau_{v,r}$: The cost of rebalancing vehicle v to request r

V_a: The set of available vehicles

t_r^r: The request time for request r

t_r^p: The pick up time for request r

t_r^d: The drop off time for request r

t_r^{opt}: The earliest drop off time for request r

t_r^{Dij}: The shortest path travel time for request r

t_{max}^w: The max waiting time accepted by the passenger

t_{max}^d: The max delay time accepted by the passenger

$x_{v,t}$: The assignment variable between vehicle and trips

S_t: The set of trips constructed in RTV graph

v_t: The set of possible trips for vehicle v

r_t: The set of possible trips for request r

θ: A candidate split

j: The feature of a candidate split

p_m: The pivot of the feature

$Q_L(\theta)$: The left subset of data for split θ

$Q_R(\theta)$: The right subset of data for split θ

$H(X_m)$: The impurity function for training data X at node m

N_m: The number of observations at node m

U_r: The utility of the request r

β: The coefficients matrix of the features

x_r: The matrix of features

ϵ: The error term of the logistic regression

Some definitions in the tables:

TotalDis: Total occupied travel distance of all the vehicles

TotalRbDis: Total rebalancing distance of all the vehicles

AvgACR: Average acceptance rate of the passengers

AvgDiscount: Average discount given to the passengers

AvgEstCost: Average estimate cost of serving a request

AvgError: Average error of the estimate cost of the requests

AvgProfit: Average expected profit of serving a request

TotalExpProfit: Total expected profit of serving the request

APPENDIX C

TABLES AND FIGURES

LIST OF TABLES

Vehicles in system:3000, MaxWaiting:300							
Day	1	2	3	4	5	6	7
#Request	292,605	304,191	330,108	336,172	352,513	378,586	392,923
Served	222,856	238,905	242,524	240,011	234,775	231,287	236,849
ServeRate	0.761	0.785	0.734	0.714	0.666	0.624	0.602
WaitingTime(s)	129	124	127	129	132	132	134
TotalDis(km)	953,591	883,975	902,277	912,426	939,738	919,371	1,017,949
TotalRbDis(km)	178,884	162,516	140,661	143,509	143,5091	132,925	138,217
Profit($)	966,199	1,128,035	1,090,001	1,046,843	939,914	927,194	932,462

Table 1: Results of non-shared ride simulation model with 3000 vehicles

Vehicles in system:5000, MaxWaiting:300							
Day	1	2	3	4	5	6	7
#Request	292,605	304,191	330,108	336,172	352,513	378,586	392,923
Served	289,624	302,284	327,066	333,036	346,735	371,856	388,701
ServeRate	0.994	0.993	0.991	0.991	0.984	0.982	0.989
WaitingTime(s)	89.0	97.2	94.6	97.1	98.0	98.9	94.9
TotalDis(km)	1,115,495	1,111,789	1,174,965	1,209,567	1,273,494	1,343,284	1,403,643
TotalRbDis(km)	184,294	171,419	170,377	165,896	185,551	215,405	209,144
Profit($)	1,418,600	1,454,428	1,568,856	1,571,683	1,623,614	1,811,996	1,855,542

Table 2: Results of non-shared ride simulation model with 5000 vehicles

Vehicles in system:3000, MaxWaiting:300,MaxDelay:600							
Day	1	2	3	4	5	6	7
#Request	292,605	304,191	330,108	336,172	352,513	378,586	392,923
Served	281,237	287,735	315,268	320,986	335,837	358,256	376,919
Shared	265,735	271,139	297,612	303,973	318,373	339,610	357,159
ServeRate	0.961	0.946	0.955	0.955	0.952	0.946	0.959
ShareRate	0.944	0.946	0.944	0.947	0.948	0.948	0.947
WaitingTime(s)	133	134	131	132	134	128	125
DelayTime(s)	190	191	186	188	190	182	178
TotalDis(km)	771,350	740,800	790,548	812,343	855,359	899,049	936,937
TotalRbDis(km)	101,205	95,080	98,279	97,036	102,353	114,056	118,700
Profit($)	2,013,263	2,031,810	2,186,268	2,228,520	2,331,624	2,497,828	2,576,067

Table 3: Results of ride-sharing simulation model with 3000 vehicles

Vehicles in system:3000, MaxWaiting:300,MaxDelay:600							
Day	1	2	3	4	5	6	7
#Request	292,605	304,191	330,108	336,172	352,513	378,586	392,923
Served	283,463	294,209	318,199	323,624	333,424	360,792	373,669
Shared	264,144	272,785	297,011	302,884	314,530	342,031	352,744
ServeRate	0.969	0.967	0.964	0.963	0.946	0.953	0.951
ShareRate	0.932	0.927	0.933	0.936	0.943	0.948	0.944
WaitingTime(s)	126	125	126	126	128	129	123
DelayTime(s)	183	181	180	180	182	183	179
TotalDis(km)	773,360	746,025	792,270	807,732	847,741	879,361	914,532
TotalRbDis(km)	98,779	92,902	94,192	92,453	99,361	107,268	113,043
AvgACR	0.858	0.862	0.863	0.862	0.861	0.864	0.866
AvgDiscount	0.218	0.224	0.223	0.224	0.223	0.217	0.214
AvgEstCost($)	4.22	3.81	3.71	3.75	3.81	3.92	3.87
AvgError	0.238	0.247	0.245	0.241	0.240	0.235	0.233
AvgProfit($)	3.78	3.77	3.73	3.71	3.70	3.69	3.68
TotalExpProfit($)	1,106,047	1,146,800	1,231,302	1,247,198	1,304,298	1,396,982	1,445,956

Table 4: Results of ride-sharing simulation model with optimal price

Vehicles in system:3000, MaxWaiting:300,MaxDelay:600							
Day	1	2	3	4	5	6	7
#Request	292,605	304,191	330,108	336,172	352,513	378,586	392,923
Served	283,241	292,632	318,224	321,380	335,592	361,928	373,669
Shared	263,981	270,099	297,858	301,133	315,792	341,660	353,117
ServeRate	0.968	0.962	0.964	0.956	0.952	0.956	0.951
ShareRate	0.932	0.923	0.936	0.937	0.941	0.944	0.945
WaitingTime(s)	128	127	126	127	126	129	123
DelayTime(s)	183	174	178	179	178	182	176
TotalDis(km)	759,789	745,348	779,258	805,987	842,627	876,804	917,361
TotalRbDis(km)	95,368	93,878	94,023	94,749	99,023	103,562	112,358
AvgACR	0.677	0.682	0.675	0.676	0.683	0.688	0.686
AvgDiscount	0.22	0.22	0.22	0.22	0.22	0.22	0.22
AvgEstCost($)	4.18	3.73	3.74	3.75	3.80	3.88	3.85
AvgError	0.232	0.239	0.243	0.237	0.225	0.232	0.236
AvgProfit($)	2.94	2.97	2.96	3.02	3.01	3.05	3.04
TotalExpProfit($)	860,258	903,447	977,119	1,015,239	1,061,064	1,154,687	1,194,485

Table 5: Results of ride-sharing simulation model with average discount

Vehicles in system:3000, MaxWaiting:300,MaxDelay:600							
Day	1	2	3	4	5	6	7
Request	292,605	304,191	330,108	336,172	352,513	378,586	392,923
Served	282,389	291,938	317,233	322,389	334,182	360,035	372,884
Shared	262,561	270,563	296,768	301,268	315,133	340,233	351,629
ServeRate	0.965	0.960	0.961	0.959	0.948	0.953	0.949
ShareRate	0.930	0.926	0.935	0.934	0.943	0.945	0.943
WaitingTime(s)	127	126	128	126	127	128	124
DelayTime(s)	182	179	180	181	179	181	177
TotalDis(km)	761,438	743,197	781,343	803,232	836,345	868,984	901,278
TotalRbDis(km)	96,587	94,108	93,899	93,650	98,147	105,347	111,269
AvgACR	0.086	0.088	0.091	0.090	0.092	0.094	0.098
AvgDiscount	0	0	0	0	0	0	0
AvgEstCost($)	4.21	3.76	3.72	3.73	3.83	3.89	3.86
AvgError	0.235	0.243	0.241	0.239	0.221	0.236	0.238
AvgProfit($)	0.56	0.58	0.59	0.58	0.61	0.64	0.63
TotalExpProfit($)	163,858	176,430	194,764	194,980	215,033	242,295	247,541

Table 6: Results of ride-sharing simulation model with original price

LIST OF FIGURES

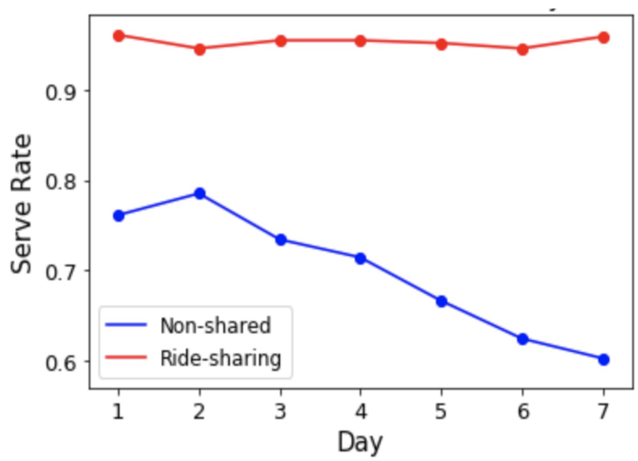

Figure 1: Serve rate of ride-sharing system and non-shared ride system

number of vehicle in both systems = 3000, max waiting = 300, max delay = 600

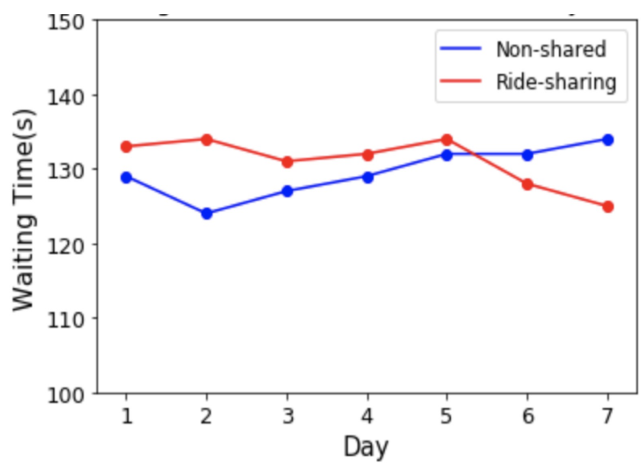

Figure 2: Average waiting time of ride-sharing system and non-shared ride system

number of vehicle in both systems = 3000, max waiting = 300, max delay = 600

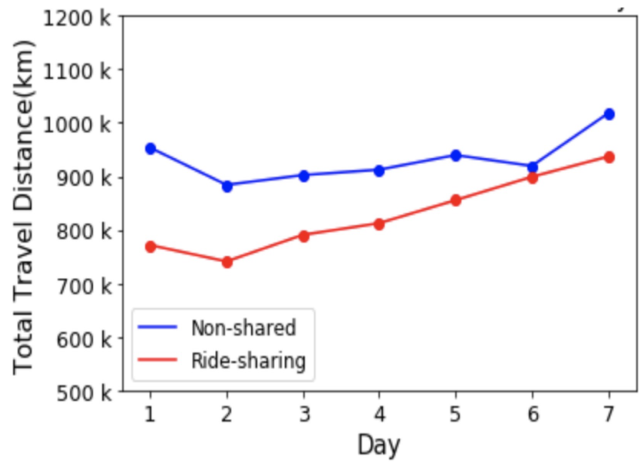

Figure 3: Total travel distance of ride-sharing system and non-shared ride system

number of vehicle in both systems = 3000, max waiting = 300, max delay = 600

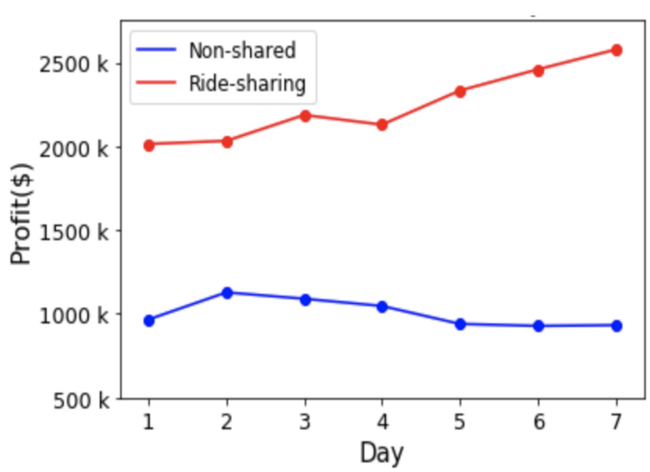

Figure 4: Total profit of ride-sharing system and non-shared ride system

number of vehicle in both systems = 3000, max waiting = 300, max delay = 600

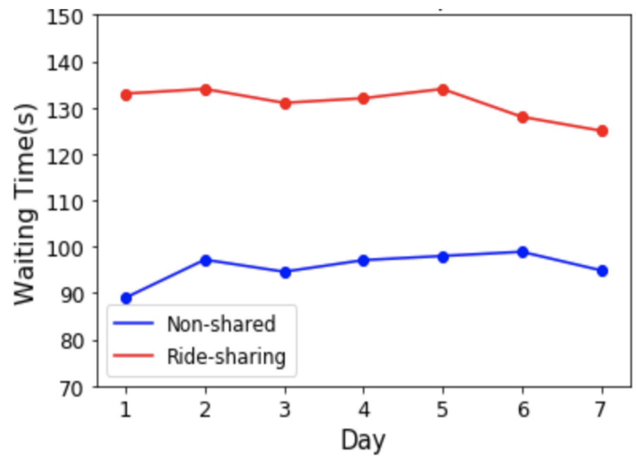

Figure 5: Average waiting time of ride-sharing system and non-shared ride system

Same scale requests are served in both systems

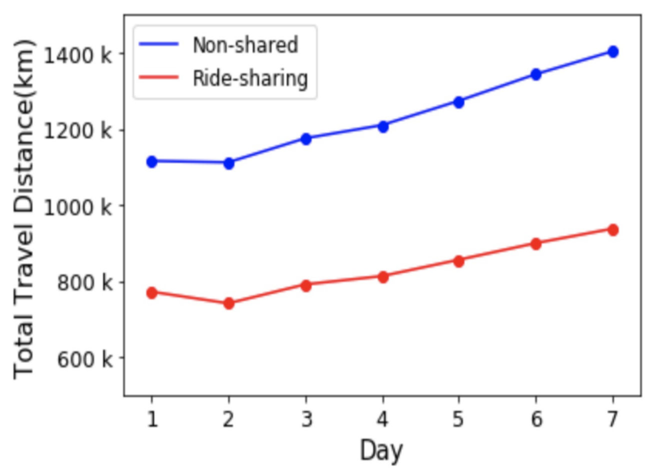

Figure 6: Total travel distance of ride-sharing system and non-shared ride system

Same scale requests are served in both systems

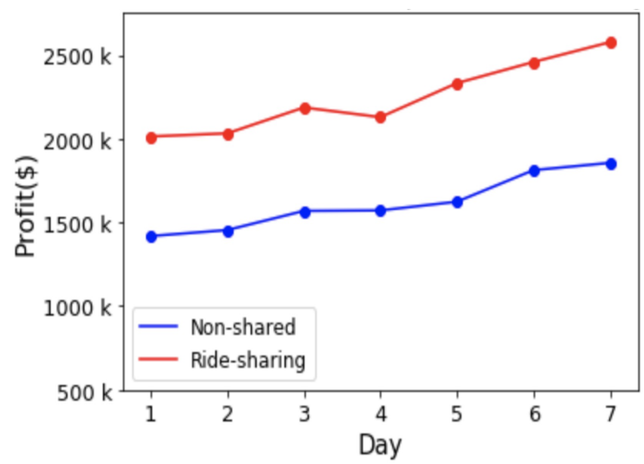

Figure 7: Total profit of ride-sharing system and non-shared ride system

Same scale requests are served in both systems

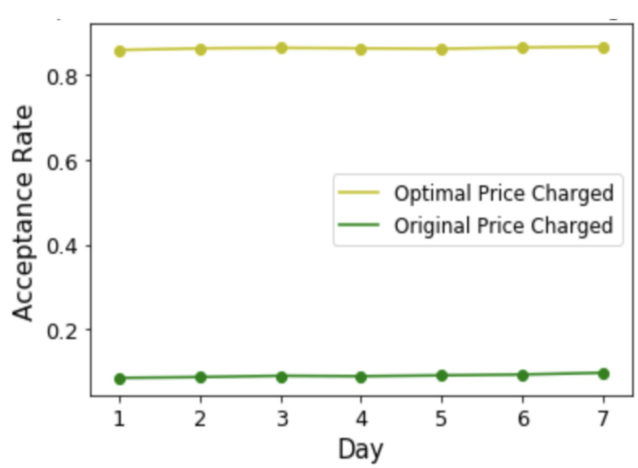

Figure 8: Average acceptance rate when different pricing strategy applied

number of vehicle = 3000, max waiting = 300, max delay = 600

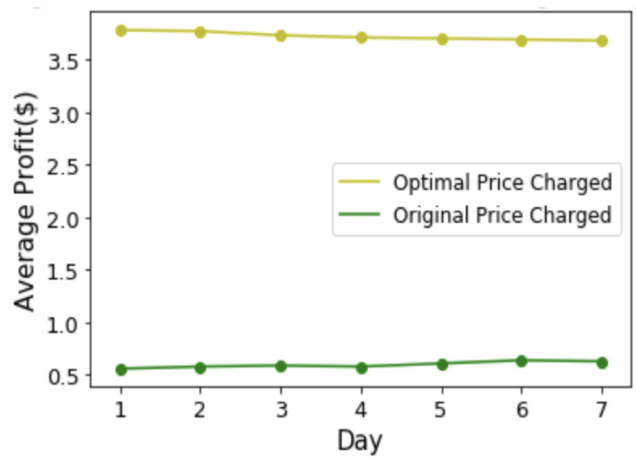

Figure 9: Average profit when different pricing strategy applied

number of vehicle = 3000, max waiting = 300, max delay = 600

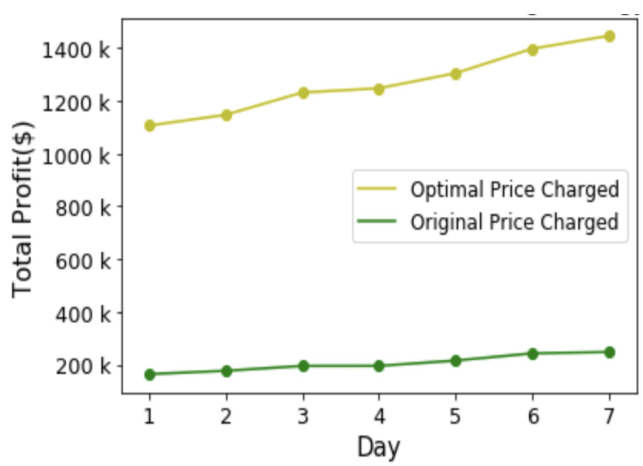

Figure 10: Average total profit when different pricing strategy applied

number of vehicle = 3000, max waiting = 300, max delay = 600

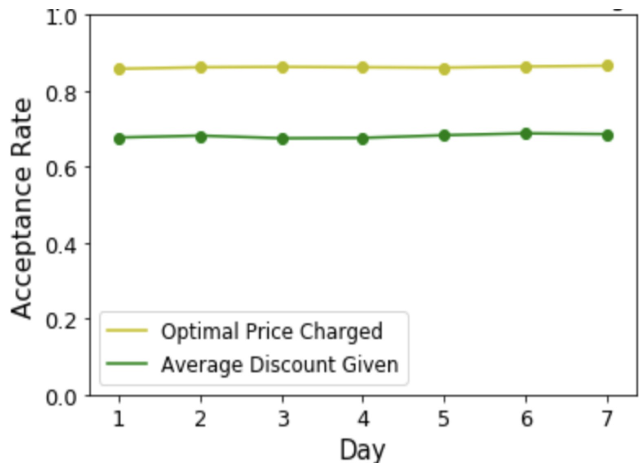

Figure 11: Average acceptance rate when different pricing strategy applied

number of vehicle = 3000, max waiting = 300, max delay = 600

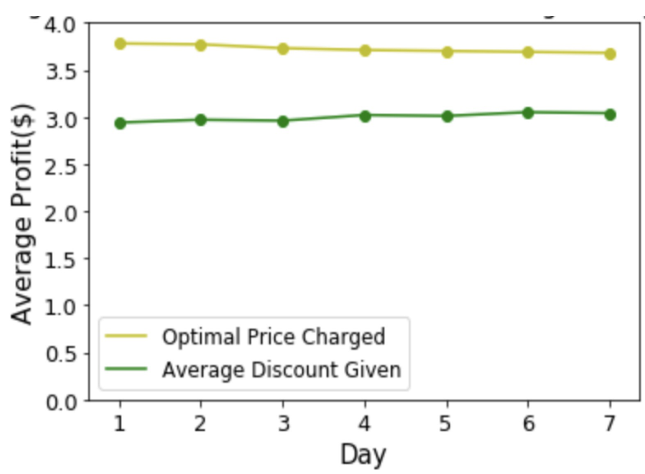

Figure 12: Average profit when different pricing strategy applied

number of vehicle = 3000, max waiting = 300, max delay = 600

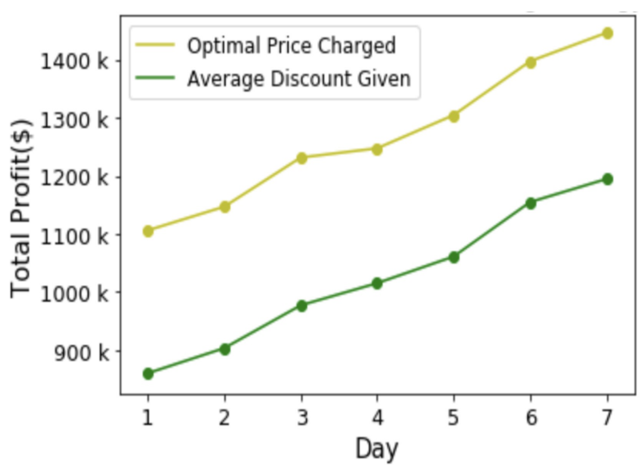

Figure 13: Average total profit when different pricing strategy applied

number of vehicle = 3000, max waiting = 300, max delay = 600